**資訊工程學系**

**114學年度專題製作期初報告**

**題目：Linux 系統自動化備份與監控**

指導老師： 教授

組長:高恩在  
組員:

**中華民國114年3月29日**

目錄

一、 摘要

二、 研究背景與目的

三、 專題內容

四、 研究方法

五、 流程圖

六、 預計完成之工作項目及具體成果

七、 工作分配與預定進度甘梯圖

八、 遭遇問題與解決方案

九、 儀器設備需求表

十、 參考資料

一、摘要

本研究專題旨在建立一個完整的自動化備份與管理系統。此系統不僅能夠自動進行資料備份，還能定期清除過期的檔案，並且管理備份過程中的日誌。通過使用 Linux 基礎命令、Shell 腳本、Docker 容器化技術、systemd 服務管理等多種技術，實現了一個功能完備且高效的備份系統。此外，學習過程還強調了錯誤檢測與通知、自動重啟機制等高可用性設計，這些技術的運用為後續滲透測試及資訊安全實踐提供了扎實的基礎。最終，本研究提供了一個可一鍵部署的腳本（deploy.sh），實現了從備份、清理到通知的全自動運行，並且可以輕鬆地移植到其他系統上運行。

二、研究背景與目的

在進入更高階的滲透測試與資訊安全領域前，學習並掌握 Linux 系統的基本操作與管理技術是至關重要的。Kali Linux 是一款廣泛應用於滲透測試的操作系統，其內建大量的安全工具與功能，因此成為了資訊安全領域專業人士的必備工具。然而，Kali Linux 的學習需要有穩固的 Linux 基礎作為支撐。

因此，本研究的目的是通過學習 Ubuntu 作為 Linux 系統的基礎，打好操作系統管理、指令行操作、腳本撰寫、容器化管理等方面的基礎知識，為日後學習 Kali Linux 及進行滲透測試等高階應用奠定穩固基礎。具體來說，本專題的研究重點包括：

* 學習 Ubuntu 操作系統的基礎知識，理解其文件系統、用戶管理、權限控制等基本操作。
* 掌握常用的系統管理工具與指令（如 Docker、systemd、cron、logrotate），並將其應用於自動化備份系統的構建。
* 了解如何在 Linux 系統中進行自動化排程、錯誤通知、日誌管理等操作，為未來學習 Kali Linux 中的自動化攻擊與防禦技術做好準備。

通過這一系列的學習與實作，本研究將為後續進行滲透測試與資訊安全實踐提供更加穩固的基礎，並確保學習者能夠順利過渡到 Kali Linux 的學習與應用。

三、專題內容

本專題的核心是設計並實現一個自動化備份與管理系統，這個系統的建構分為以下幾個重要步驟：

1. Linux 基礎命令學習

我們從學習基礎命令開始，包括如何使用 mkdir 創建資料夾、chmod 控制權限、tar 打包備份檔案、以及如何使用 find 刪除過期檔案等基本操作。

1. Shell 腳本編寫與 Systemd 設定

撰寫自動化備份腳本 backup.sh，並利用 systemd 管理備份服務與定時排程。系統會自動備份資料夾中的檔案並生成 .tar.gz 檔案，同時清除 7 天前的過期檔案。

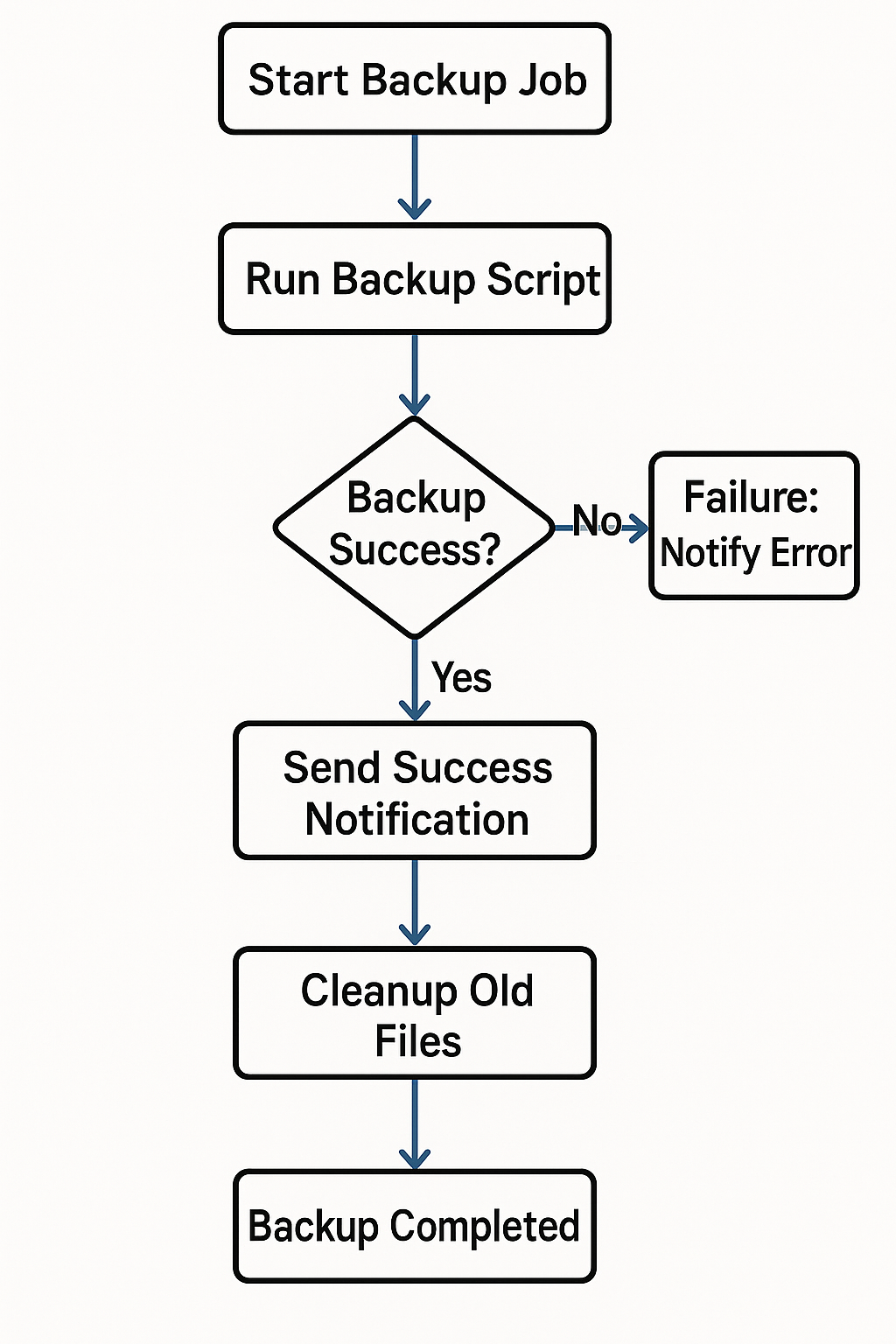
1. Docker 容器化技術  
   使用 Docker 將備份邏輯封裝成容器，實現自動備份的可移植性與可複製性。這樣可以輕鬆將備份流程從一台主機轉移到另一台，且不會受到操作系統或環境差異的影響。
2. 日誌與過期檔案管理  
   使用 logrotate 管理備份過程中的日誌檔案，並利用 find 命令清除過期的備份檔案，以減少硬碟空間的佔用。
3. 錯誤處理與重啟機制  
   使用 systemd 配置服務，並設定自動重啟機制，當備份失敗時系統會自動重啟服務並重新嘗試。
4. 一鍵部署腳本  
   撰寫 deploy.sh 部署腳本，能夠一鍵安裝並自動配置所有的備份、排程服務、Docker 容器及錯誤處理機制。這樣即使是新手，也能夠輕鬆地將整個系統部署並運行。

四、研究方法

1. Linux 基礎命令與檔案管理  
   學會了如何在 Linux 上進行基本的檔案與資料夾管理，包括使用 mkdir 創建資料夾、chmod 修改檔案權限、tar 打包與解壓縮備份檔案、以及使用 find 指令刪除過期檔案。
2. Shell 腳本編寫與自動化管理  
   利用 Bash 撰寫了自動化備份腳本，通過條件語句（if）、迴圈（for）及函數（function）實現了備份、清理與通知等功能，並結合 systemd 實現了服務管理與定時排程。
3. Docker 技術應用  
   利用 Dockerfile 將備份腳本封裝成 Docker 容器，確保無論在哪個環境中，都能一致地運行備份腳本，並且能夠通過 Docker Compose 進行一鍵部署。
4. 系統服務與排程設定  
   使用 systemd 配置了備份服務，並設置了定時排程，使得備份能夠在每日固定時間自動執行，並能在失敗時自動重試。
5. 報告與錯誤處理  
   使用 journalctl 查看備份服務的日誌，並且對錯誤進行處理，若備份失敗，會自動重啟服務並發送錯誤通知。

五、流程圖

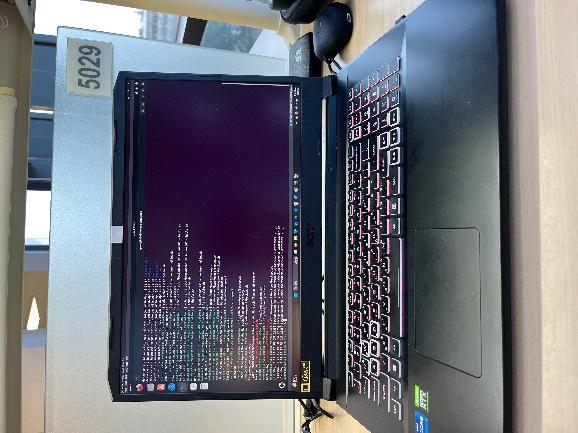
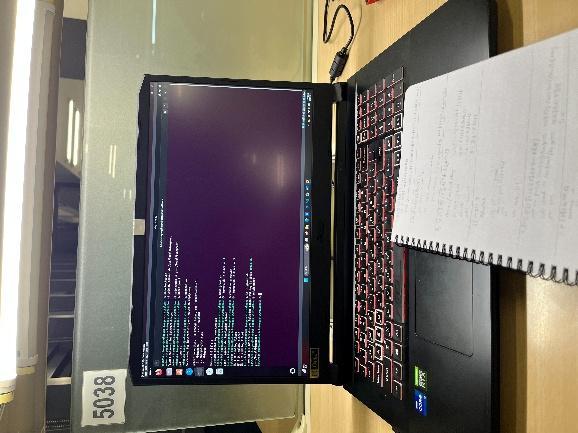
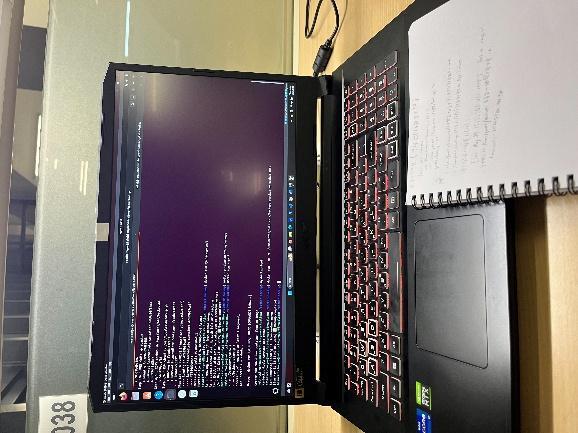
以下是自動化備份系統的流程圖：

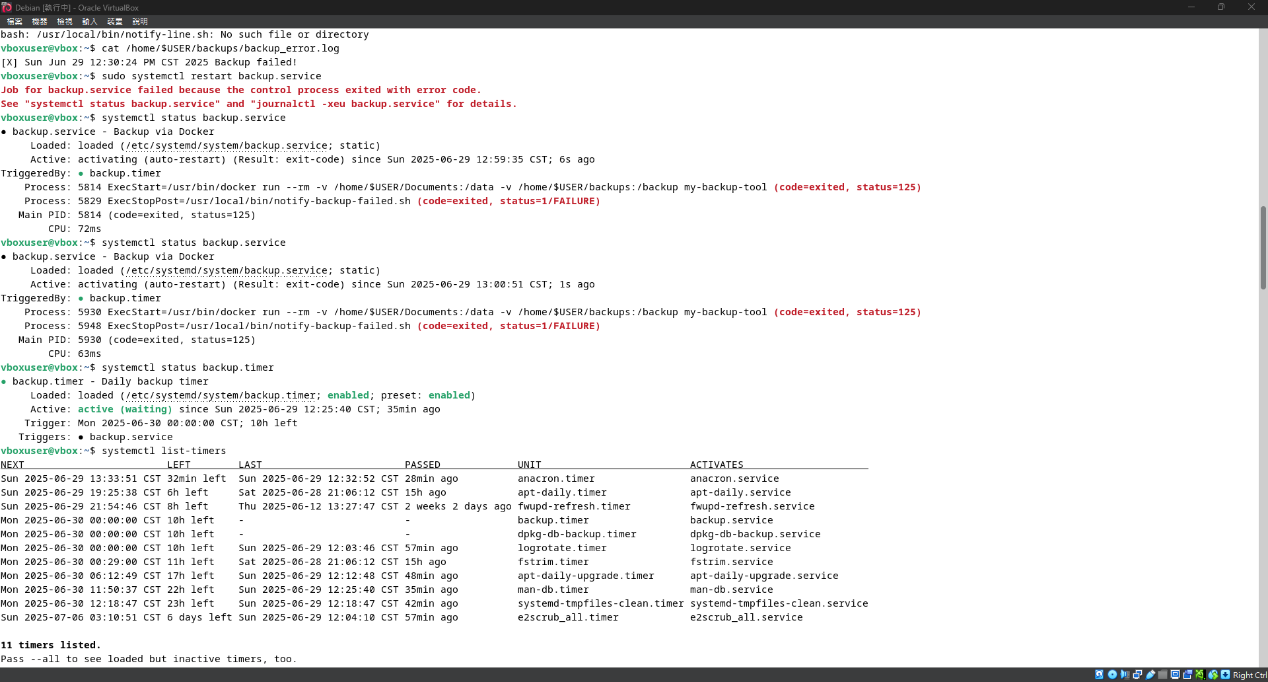
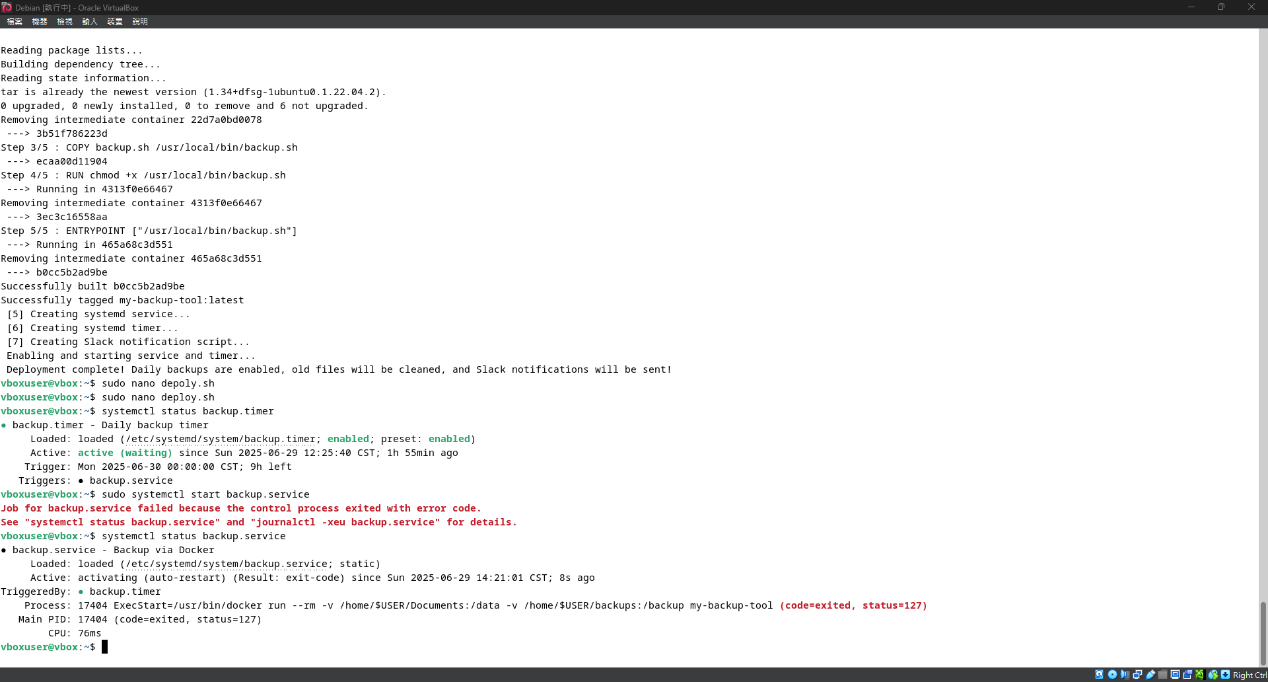
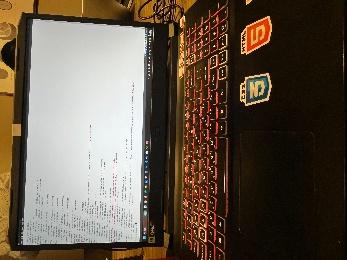
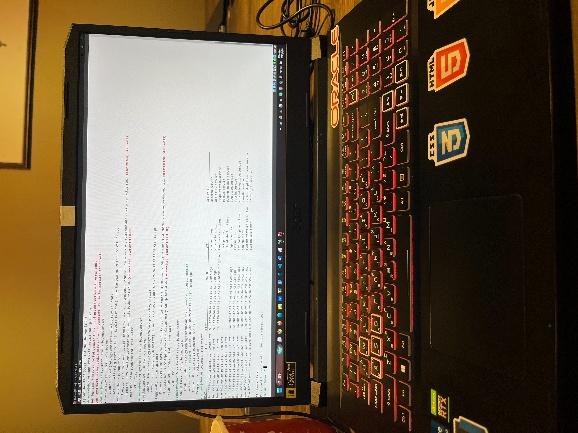
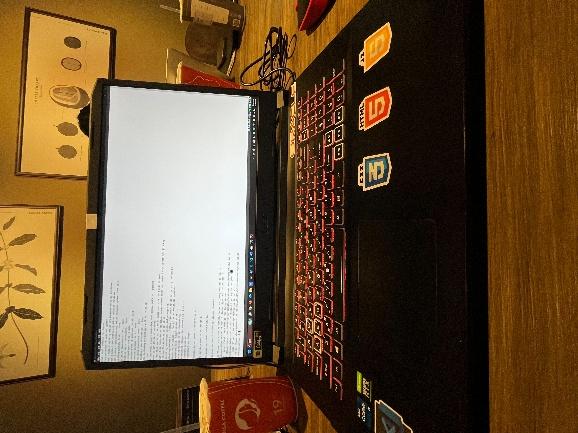


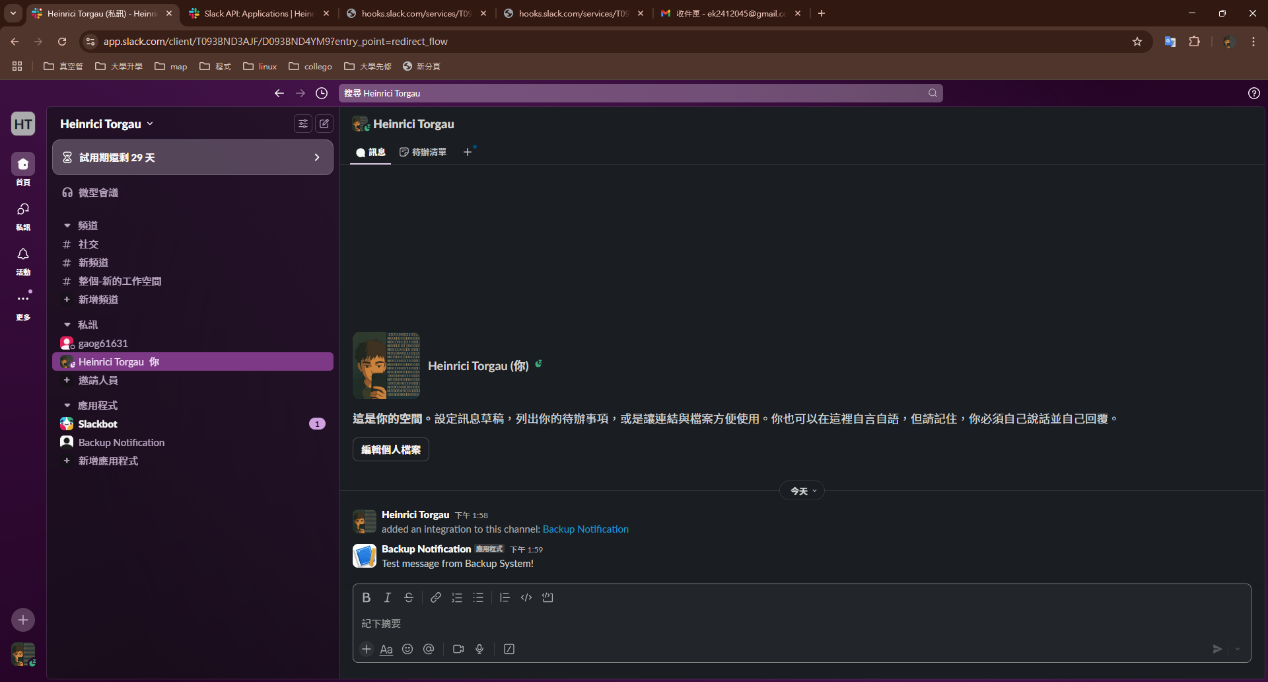
六、預計完成之工作項目及具體成果

| 項目 | 預計成果 |
| --- | --- |
| 備份腳本撰寫 | 自動化備份並生成壓縮檔案 |
| systemd 服務設置 | 完成備份服務與定時排程設置 |
| 日誌與清理機制 | 完成 logrotate 與過期檔案清理 |
| 一鍵部署腳本 | deploy.sh 完整自動化部署 |

實驗圖片







補充腳本:

**1. backup.sh：tar 備份 + 清除過期 + 發送通知**

這是執行備份的主腳本，它會將指定目錄的資料壓縮並儲存，並在備份成功後發送通知。它還會清理過期的備份檔案。

#!/bin/bash

# 設定日期，備份檔案名稱

TODAY=$(date +%Y%m%d)

SRC="/data" # 要備份的資料夾

DEST="/backup" # 備份目標資料夾

FILENAME="documents-$TODAY.tar.gz" # 備份檔案名稱

# 創建目標資料夾

mkdir -p "$DEST"

# 執行 tar 備份

tar -czf "$DEST/$FILENAME" "$SRC"

# 清除7天前的備份檔案

find "$DEST" -name "documents-\*.tar.gz" -mtime +7 -delete

# 顯示備份完成的訊息

echo "[$(date)] V Backup completed: $FILENAME"

# 發送通知（這裡用 Slack 通知作為範例）

/usr/local/bin/notify-slack.sh "V Backup completed: $FILENAME"

* **tar**：將資料夾 /data 壓縮為 .tar.gz 格式。
* **find**：刪除 7 天前的過期備份檔案。
* **通知**：使用 notify-slack.sh 發送成功通知。

**2. deploy.sh：一鍵部署備份服務與定時器**

這是用來自動化部署備份服務與定時器的腳本。當你執行這個腳本時，它會自動完成所有的配置，包括備份腳本、Docker 映像的創建，並且設置 systemd 服務與定時器。

#!/bin/bash

set -e # 如果有錯誤就停止執行

echo " [1] 創建目錄結構..."

mkdir -p ~/Documents ~/backups ~/docker-backup

cd ~/docker-backup

echo " [2] 創建備份腳本..."

cat > backup.sh << 'EOF'

#!/bin/bash

TODAY=$(date +%Y%m%d)

SRC="/data"

DEST="/backup"

FILENAME="documents-$TODAY.tar.gz"

mkdir -p "$DEST"

tar -czf "$DEST/$FILENAME" "$SRC"

find "$DEST" -name "documents-\*.tar.gz" -mtime +7 -delete

echo "[$(date)] V Backup completed: $FILENAME"

# Slack 通知（可選）

/usr/local/bin/notify-slack.sh "V Backup completed: $FILENAME"

EOF

chmod +x backup.sh

echo " [3] 創建 Dockerfile..."

cat > Dockerfile << 'EOF'

FROM ubuntu:22.04

RUN apt update && apt install -y tar

COPY backup.sh /usr/local/bin/backup.sh

RUN chmod +x /usr/local/bin/backup.sh

ENTRYPOINT ["/usr/local/bin/backup.sh"]

EOF

echo " [4] 創建 Docker 映像..."

docker build -t my-backup-tool .

echo " [5] 創建 systemd 服務..."

sudo tee /etc/systemd/system/backup.service > /dev/null << 'EOF'

[Unit]

Description=Backup via Docker

After=network.target docker.service

Wants=docker.service

[Service]

Type=oneshot

ExecStart=/usr/bin/docker run --rm \

-v /home/$USER/Documents:/data \

-v /home/$USER/backups:/backup \

my-backup-tool

Restart=on-failure

RestartSec=10

EOF

echo " [6] 創建 systemd timer..."

sudo tee /etc/systemd/system/backup.timer > /dev/null << 'EOF'

[Unit]

Description=Daily backup timer

[Timer]

OnCalendar=daily

Persistent=true

[Install]

WantedBy=timers.target

EOF

echo " [7] 創建 Slack 通知腳本..."

sudo tee /usr/local/bin/notify-slack.sh > /dev/null << 'EOF'

#!/bin/bash

WEBHOOK\_URL="YOUR\_SLACK\_WEBHOOK\_URL" # 在這裡填上你的 Webhook URL

MESSAGE="$1"

curl -X POST -H 'Content-type: application/json' \

--data "{\"text\":\"${MESSAGE}\"}" \

$WEBHOOK\_URL

EOF

sudo chmod +x /usr/local/bin/notify-slack.sh

echo "V 啟用並啟動服務與排程..."

sudo systemctl daemon-reexec

sudo systemctl enable --now backup.timer

echo " 部署完成！每天自動備份、清除舊檔，並會發送 Slack 通知！"

* **backup.sh**：自動進行備份並發送通知。
* **Dockerfile**：創建 Docker 映像來封裝備份腳本。
* **backup.service & backup.timer**：systemd 服務與定時器，用來定時執行備份。
* **notify-slack.sh**：用來發送 Slack 通知。

**3. backup.service 和 backup.timer（systemd）**

**backup.service**

這是 systemd 服務檔案，它負責執行備份腳本，並確保服務出錯後會自動重啟。

[Unit]

Description=Backup via Docker

After=network.target docker.service

Wants=docker.service

[Service]

Type=oneshot

ExecStart=/usr/bin/docker run --rm \

-v /home/$USER/Documents:/data \

-v /home/$USER/backups:/backup \

my-backup-tool

Restart=on-failure

RestartSec=10

* **ExecStart**：運行 Docker 容器並執行備份。
* **Restart=on-failure**：當備份失敗時自動重啟服務。

**backup.timer**

這是 systemd 定時器，負責設置定期執行備份（例如每天一次）。

[Unit]

Description=Daily backup timer

[Timer]

OnCalendar=daily

Persistent=true

[Install]

WantedBy=timers.target

* **OnCalendar=daily**：定時觸發備份，這裡設置為每天。
* **Persistent=true**：如果系統停機，會保證下一次開機時仍能執行。

七、工作分配與預定進度甘梯圖

| 週次 | 任務 | 完成進度 |
| --- | --- | --- |
| Day 1 ~ 5 | Linux 基礎與排程 | 已完成 |
| Day 6 ~ 10 | Shell Script 與 Systemd | 已完成 |
| Day 11 ~ 13 | Docker 備份容器與自動化排程 | 已完成 |
| Day 14 ~ 15 | 自動清除過期備份與 log 管理 | 已完成 |
| Day 16 | Docker Compose 一鍵啟動備份系統 | 已完成 |
| Day 17 | 整合報表與系統結構 | 已完成 |
| Day 18 | 異常監控與自動重啟 | 已完成 |
| Day 19 | deploy.sh 一鍵部署整合腳本 | 已完成 |

八、遭遇問題與解決方案

1. Docker 容器掛載資料夾問題：
   * 問題：初期使用者資料夾無法正確掛載到容器內。
   * 解決方案：確認容器內的掛載路徑與主機路徑一致，並設定適當的資料夾權限。
2. systemd 定時器無法觸發：
   * 問題：backup.timer 在設定後未觸發備份任務。
   * 解決方案：重新加載 systemd 設定並檢查錯誤日誌，調整排程設置。

九、儀器設備需求表

| 設備 | 用途 | 數量 |
| --- | --- | --- |
| 電腦伺服器（Ubuntu） | 執行備份系統，管理資料 | 1 |
| Docker 容器化環境 | 用來執行備份腳本 | 1 |
| 網路連線 | 用來發送 LINE Notify 通知 | 1 |
| 備份資料磁碟 | 儲存備份檔案 | 1 |

十、參考資料

1. Docker 官方文檔：<https://docs.docker.com/>
2. systemd 官方文檔： <https://www.freedesktop.org/wiki/Software/systemd/>
3. logrotate 官方文檔：<https://linux.die.net/man/8/logrotate>